Practice On Student Details

September 29, 2025

[1]:
```
# importing pandas as pd for using data frame
import pandas as pd

# creating dataframe with student details
dataframe = pd.DataFrame({'id': [7058, 4511, 7014, 7033],
                          'name': ['sravan', 'manoj', 'aditya', 'bhanu'],
                          'Maths_marks': [99, 97, 88, 90],
                          'Chemistry_marks': [89, 99, 99, 90],
                          'telugu_marks': [99, 97, 88, 80],
                          'hindi_marks': [99, 97, 56, 67],
                          'social_marks': [79, 97, 78, 90], })

# display dataframe
dataframe
```

[1]:

	id	name	Maths_marks	Chemistry_marks	telugu_marks	hindi_marks	social_marks
0	7058	sravan	99	89	99	99	79
1	4511	manoj	97	99	97	97	97
2	7014	aditya	88	99	88	56	78
3	7033	bhanu	90	90	80	67	90

[2]:
```
# describing the data frame
print(dataframe.describe())
```

```
                id  Maths_marks  Chemistry_marks  telugu_marks  hindi_marks  \
count     4.000000     4.000000             4.00       4.00000     4.000000
mean   6404.000000    93.500000            94.25      91.00000    79.750000
std    1262.128625     5.322906             5.50       8.75595    21.561926
min    4511.000000    88.000000            89.00      80.00000    56.000000
25%    6388.250000    89.500000            89.75      86.00000    64.250000
50%    7023.500000    93.500000            94.50      92.50000    82.000000
75%    7039.250000    97.500000            99.00      97.50000    97.500000
```

```
max    7058.000000    99.000000            99.00      99.00000    99.000000
```

```
       social_marks
count      4.000000
mean      86.000000
std        9.128709
min       78.000000
25%       78.750000
50%       84.500000
75%       91.750000
max       97.000000
```

[3]:
```
# finding unique values
print(dataframe['Maths_marks'].unique())
```

```
[99 97 88 90]
```

[4]:
```
# counting unique values
print(dataframe['Maths_marks'].nunique())
```

```
4
```

[5]:
```
# display the columns in the data frame
print(dataframe.columns)
```

```
Index(['id', 'name', 'Maths_marks', 'Chemistry_marks', 'telugu_marks',
       'hindi_marks', 'social_marks'],
      dtype='object')
```

[6]:
```
# information about dataframe
print(dataframe.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               4 non-null      int64
 1   name             4 non-null      object
 2   Maths_marks      4 non-null      int64
 3   Chemistry_marks  4 non-null      int64
 4   telugu_marks     4 non-null      int64
 5   hindi_marks      4 non-null      int64
 6   social_marks     4 non-null      int64
dtypes: int64(6), object(1)
memory usage: 356.0+ bytes
None
```

[7]:
```
# getting all minimum values from all columns in a dataframe
print(dataframe.min())
```

```
id                  4511
name              aditya
Maths_marks           88
Chemistry_marks       89
telugu_marks          80
hindi_marks           56
social_marks          78
dtype: object
```

[8]:
```
# minimum value from a particular column in a data frame
print(dataframe['Maths_marks'].min())
```

```
88
```

[9]:
```
# computing maximum values
print(dataframe.max())
```

```
id                  7058
name              sravan
Maths_marks           99
Chemistry_marks       99
telugu_marks          99
hindi_marks           99
social_marks          97
dtype: object
```

[11]:
```
# computing sum of each column separately
print(dataframe.sum())
```

```
id                                25616
name              sravanmanojadityabhanu
Maths_marks                          374
Chemistry_marks                      377
telugu_marks                         364
hindi_marks                          319
social_marks                         344
dtype: object
```

[12]:
```
# finding count of cells in each column separately
print(dataframe.count())
```

```
id                4
name              4
Maths_marks       4
Chemistry_marks   4
telugu_marks      4
hindi_marks       4
social_marks      4
dtype: int64
```

[14]:
```
# Select only numeric columns
numeric_df = dataframe.select_dtypes(include='number')

# Now compute standard deviation for numeric columns only
print("Standard deviation of numeric columns:")
print(numeric_df.std())
```

```
Standard deviation of numeric columns:
id                1262.128625
Maths_marks          5.322906
Chemistry_marks      5.500000
telugu_marks         8.755950
hindi_marks         21.561926
social_marks         9.128709
dtype: float64
```

[15]:
```
# Computing variance for numeric columns only
print("Standard deviation of numeric columns:")
print(numeric_df.var())
```

```
Standard deviation of numeric columns:
id                1.592969e+06
Maths_marks       2.833333e+01
Chemistry_marks   3.025000e+01
telugu_marks      7.666667e+01
hindi_marks       4.649167e+02
social_marks      8.333333e+01
dtype: float64
```

[16]:
```
# group by name
print(dataframe.groupby('name').first())
```

```
          id  Maths_marks  Chemistry_marks  telugu_marks  hindi_marks  \
name
aditya  7014           88               99            88           56
bhanu   7033           90               90            80           67
manoj   4511           97               99            97           97
sravan  7058           99               89            99           99

        social_marks
name
aditya            78
bhanu             90
manoj             97
sravan            79
```

[17]:
```
# group by name with social_marks sum
print(dataframe.groupby('name')['social_marks'].sum())
```

```
name
aditya    78
bhanu     90
manoj     97
sravan    79
Name: social_marks, dtype: int64
```

[18]:
```
# group by name with maths_marks count
print(dataframe.groupby('name')['Maths_marks'].count())
```

```
name
aditya    1
bhanu     1
manoj     1
sravan    1
Name: Maths_marks, dtype: int64
```

[19]:
```
# group by name with maths_marks
print(dataframe.groupby('name')['Maths_marks'])
```

```
<pandas.core.groupby.generic.SeriesGroupBy object at 0x0000014A6C1DF6F0>
```

[21]:
```
dataframe.head()      # first 5 rows
```

[21]:

	id	name	Maths_marks	Chemistry_marks	telugu_marks	hindi_marks	social_marks
0	7058	sravan	99	89	99	99	79
1	4511	manoj	97	99	97	97	97
2	7014	aditya	88	99	88	56	78
3	7033	bhanu	90	90	80	67	90

[22]:
```
dataframe.tail(3)     # last 3 rows
```

[22]:

	id	name	Maths_marks	Chemistry_marks	telugu_marks	hindi_marks	social_marks
1	4511	manoj	97	99	97	97	97
2	7014	aditya	88	99	88	56	78
3	7033	bhanu	90	90	80	67	90

[23]:
```
dataframe.shape       # (number of rows, number of columns)
```

[23]:
```
(4, 7)
```

[24]:
```
dataframe['Maths_marks']        # select a single column
```

[24]:
```
0    99
1    97
2    88
3    90
Name: Maths_marks, dtype: int64
```

[25]:
```
dataframe[['Maths_marks','hindi_marks']]  # select multiple columns
```

[25]:
```
   Maths_marks  hindi_marks
0           99           99
1           97           97
2           88           56
3           90           67
```

[26]:
```
dataframe.iloc[0]               # select first row by index
```

[26]:
```
id                   7058
name                sravan
Maths_marks             99
Chemistry_marks         89
telugu_marks            99
hindi_marks             99
social_marks            79
Name: 0, dtype: object
```

[27]:
```
dataframe.loc[0,'Maths_marks']  # select specific cell
```

[27]:
```
np.int64(99)
```

[28]:
```
# Students with Maths_marks > 90
dataframe[dataframe['Maths_marks'] > 90]
```

[28]:
```
     id    name  Maths_marks  Chemistry_marks  telugu_marks  hindi_marks  \
0  7058  sravan           99               89            99           99
1  4511   manoj           97               99            97           97

   social_marks
0            79
1            97
```

[29]:
```
# Students with Maths_marks > 90 AND Hindi_marks > 80
dataframe[(dataframe['Maths_marks'] > 90) & (dataframe['hindi_marks'] > 80)]
```

[29]:
```
     id    name  Maths_marks  Chemistry_marks  telugu_marks  hindi_marks  \
0  7058  sravan           99               89            99           99
1  4511   manoj           97               99            97           97

   social_marks
0            79
1            97
```

[32]:
```
# Select numeric columns only
numeric_df = dataframe.select_dtypes(include='number')

# Print aggregations nicely
print("===== Minimum Values =====")
print(numeric_df.min())
print("\n===== Maximum Values =====")
print(numeric_df.max())
print("\n===== Sum =====")
print(numeric_df.sum())
print("\n===== Mean =====")
print(numeric_df.mean())
print("\n===== Standard Deviation =====")
print(numeric_df.std())
print("\n===== Variance =====")
print(numeric_df.var())
print("\n===== Count of Non-Null Values =====")
print(numeric_df.count())
```

```
===== Minimum Values =====
id                 4511
Maths_marks          88
Chemistry_marks      89
telugu_marks         80
hindi_marks          56
social_marks         78
dtype: int64

===== Maximum Values =====
id                 7058
Maths_marks          99
Chemistry_marks      99
telugu_marks         99
hindi_marks          99
social_marks         97
dtype: int64

===== Sum =====
id                 25616
Maths_marks          374
```

```
Chemistry_marks      377
telugu_marks         364
hindi_marks          319
social_marks         344
dtype: int64

===== Mean =====
id                 6404.00
Maths_marks          93.50
Chemistry_marks      94.25
telugu_marks         91.00
hindi_marks          79.75
social_marks         86.00
dtype: float64

===== Standard Deviation =====
id                 1262.128625
Maths_marks           5.322906
Chemistry_marks       5.500000
telugu_marks          8.755950
hindi_marks          21.561926
social_marks          9.128709
dtype: float64

===== Variance =====
id                 1.592969e+06
Maths_marks        2.833333e+01
Chemistry_marks    3.025000e+01
telugu_marks       7.666667e+01
hindi_marks        4.649167e+02
social_marks       8.333333e+01
dtype: float64

===== Count of Non-Null Values =====
id                 4
Maths_marks        4
Chemistry_marks    4
telugu_marks       4
hindi_marks        4
social_marks       4
dtype: int64
```

[33]:
```
dataframe['name'].unique()    # all unique names
```

[33]:
```
array(['sravan', 'manoj', 'aditya', 'bhanu'], dtype=object)
```

[34]:
```
dataframe['name'].nunique()   # count of unique names
```

[34]:
```
4
```

[35]:
```
# Group by 'name' and get first row of each group
dataframe.groupby('name').first()
```

[35]:
```
          id  Maths_marks  Chemistry_marks  telugu_marks  hindi_marks  \
name
aditya  7014           88               99            88           56
bhanu   7033           90               90            80           67
manoj   4511           97               99            97           97
sravan  7058           99               89            99           99

        social_marks
name
aditya            78
bhanu             90
manoj             97
sravan            79
```

[36]:
```
# Group by 'name' and sum social_marks
dataframe.groupby('name')['social_marks'].sum()
```

[36]:
```
name
aditya    78
bhanu     90
manoj     97
sravan    79
Name: social_marks, dtype: int64
```

[38]:
```
dataframe.sort_values(by='Maths_marks')            # ascending
```

[38]:
```
     id    name  Maths_marks  Chemistry_marks  telugu_marks  hindi_marks  \
2  7014  aditya           88               99            88           56
3  7033   bhanu           90               90            80           67
1  4511   manoj           97               99            97           97
0  7058  sravan           99               89            99           99

   social_marks
2            78
3            90
1            97
0            79
```

[39]:
```
dataframe.sort_values(by='Maths_marks', ascending=False)  # descending
```

[39]:
```
     id    name  Maths_marks  Chemistry_marks  telugu_marks  hindi_marks  \
0  7058  sravan           99               89            99           99
1  4511   manoj           97               99            97           97
```

```
3  7033   bhanu          90               90            80           67
2  7014  aditya          88               99            88           56

   social_marks
0            79
1            97
3            90
2            78
```

[40]:
```
# Add a new column for total marks (Maths + Chemistry + Hindi marks)
dataframe['Total_marks'] = dataframe['Maths_marks'] +␣
 ↪dataframe['Chemistry_marks'] + dataframe['hindi_marks']
```

[41]:
```
dataframe
```

[41]:
```
     id    name  Maths_marks  Chemistry_marks  telugu_marks  hindi_marks  \
0  7058  sravan           99               89            99           99
1  4511   manoj           97               99            97           97
2  7014  aditya           88               99            88           56
3  7033   bhanu           90               90            80           67

   social_marks  Total_marks
0            79          287
1            97          293
2            78          243
3            90          247
```

[42]:
```
# Add bonus column (10% of Maths_marks)
dataframe['Bonus'] = 0.1 * dataframe['Maths_marks']
```

[47]:
```
# Display only Maths_marks and Bonus columns
print(dataframe[['Maths_marks', 'Bonus']])
```

```
   Maths_marks  Bonus
0           99    9.9
1           97    9.7
2           88    8.8
3           90    9.0
```

[44]:
```
dataframe.isnull()           # check for missing values
```

[44]:
```
      id   name  Maths_marks  Chemistry_marks  telugu_marks  hindi_marks  \
0  False  False        False            False         False        False
1  False  False        False            False         False        False
2  False  False        False            False         False        False
3  False  False        False            False         False        False

   social_marks  Total_marks  Bonus
```

```
0         False        False  False
1         False        False  False
2         False        False  False
3         False        False  False
```

[45]:
```
dataframe.dropna()           # drop rows with missing values
```

[45]:
```
     id    name  Maths_marks  Chemistry_marks  telugu_marks  hindi_marks  \
0  7058  sravan           99               89            99           99
1  4511   manoj           97               99            97           97
2  7014  aditya           88               99            88           56
3  7033   bhanu           90               90            80           67

   social_marks  Total_marks  Bonus
0            79          287    9.9
1            97          293    9.7
2            78          243    8.8
3            90          247    9.0
```

[46]:
```
dataframe.fillna(0)          # fill missing values with 0
```

[46]:
```
     id    name  Maths_marks  Chemistry_marks  telugu_marks  hindi_marks  \
0  7058  sravan           99               89            99           99
1  4511   manoj           97               99            97           97
2  7014  aditya           88               99            88           56
3  7033   bhanu           90               90            80           67

   social_marks  Total_marks  Bonus
0            79          287    9.9
1            97          293    9.7
2            78          243    8.8
3            90          247    9.0
```

[48]:
```
dataframe.to_csv('students_cleaned.csv', index=False)  # save as CSV
```

[49]:
```
dataframe.to_excel('students_cleaned.xlsx', index=False)  # save as Excel
```

[ ]:
```
!jupyter nbconvert --to pdf "Practice On Student Details.ipynb" --output "C:/
↪Users/ASUS/Downloads/practice_on_student_details.pdf"
```